

```

1 //[集合堆疊電腦/The SetStack Computer](1/3)
2 #define IN "P17IN.txt"
3 #define OUT "P17OUT.txt"
4 //*****
5 #include <iostream>
6 #include <ctime>
7 using namespace std;
8 void redir(void);
9 //*****
10 /* Work Space*/
11 #include <string>
12 #include <set>
13 #include <map>
14 #include <vector>
15 #include <stack>
16 #include <algorithm> //set_union(), set_intersection()
17 #include <iterator> //inserter()
18
19 typedef set<int> Set;
20 map<Set, int> IdCache; //Set -> ID
21 vector<Set> SetCache; //ID -> Set
22
23 #define ALL(x) x.begin(),x.end()
24 #define INS(x) inserter(x,x.begin())
25
26 int ID(Set x);
27 //*****
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

```

64 //[集合堆疊電腦/The SetStack Computer](2/3)
65 int main(void)
66 {
67     redir(); //redirection
68 //*****
69 /* Work Space*/
70     int T, n;
71     stack<int> s;//SetID
72
73
74     scanf("%d", &T);
75     while(T--){
76         scanf("%d", &n);
77         while(n--){
78             string op; //block scope
79             cin >> op;
80
81             if(op[0] == 'P'){//PUSH
82                 Set x; //block scope
83                 s.push(ID(x));
84             }else if(op[0] == 'D'){//DUP
85                 s.push(s.top());
86             }else{
87                 Set x1, x2, x; //block scope
88                 x1 = SetCache[s.top()];
89                 s.pop();
90                 x2 = SetCache[s.top()];
91                 s.pop();
92
93                 if(op[0] == 'U'){//UNION
94                     set_union(ALL(x1), ALL(x2), INS(x));
95                 }else if(op[0] == 'I'){//INTERSECT
96                     set_intersection(ALL(x1), ALL(x2), INS(x));
97                 }else{//ADD
98                     x = x2;
99                     x.insert(ID(x1));
100                 }
101
102                 s.push(ID(x));
103             }
104             cout << SetCache[s.top()].size() << endl;
105         }
106         cout << "****" << endl;
107     }
108 //*****
109     freopen("CON", "r", stdin); //取消重新導向
110     freopen("CON", "w", stdout);
111
112     printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間
113
114     system("pause");
115     return 0; //the end...
116 }
117
118 void redir(void)
119 {
120     freopen(IN, "r", stdin);
121     freopen(OUT, "w", stdout);
122 }
123 //*****
124
125
126

```

```
127 //[集合堆疊電腦/The SetStack Computer](3/3)
128 /* Work Space*/
129 int ID(Set x)
130 {
131     if(IdCache.count(x)){
132         return IdCache[x];
133     }
134     SetCache.push_back(x);
135     return IdCache[x] = SetCache.size()-1;
136 }
137
138 //Input(IN) Sample
139 /*
140 2
141 9
142 PUSH
143 DUP
144 ADD
145 PUSH
146 ADD
147 DUP
148 ADD
149 DUP
150 UNION
151 5
152 PUSH
153 PUSH
154 ADD
155 PUSH
156 INTERSECT
157 */
158
159 //Output(OUT)
160 /*
161 0
162 0
163 1
164 0
165 1
166 1
167 2
168 2
169 2
170 ***
171 0
172 0
173 1
174 0
175 0
176 ***
177 */
```